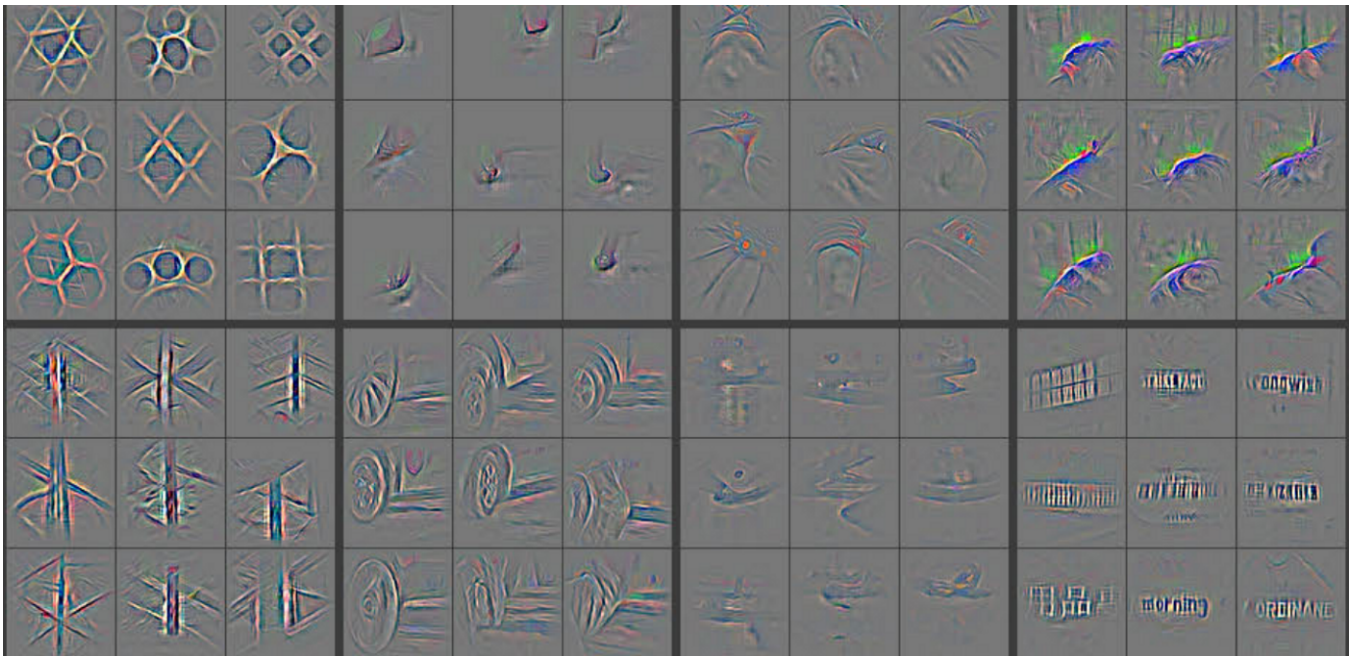
Loss functions based on feature activation and style loss.

Christopher Thomas BSc Hons. MIAP
Mar 14, 2019 · 5 min read



Feature activations in Convolutional Neural Networks. Source: https://arxiv.org/pdf/1311.2901.pdf

Loss functions using these techniques can be used during the training of U-Net based model architectures and could be applied to the training of other Convolutional Neural Networks that are generating an image as their predication/output.

I've separated this out from my article on Super Resolution (https://towardsdatascience.com/deep-learning-based-super-resolution-without-using-a-gan-11c9bb5b6cd5), to be more generic as I am using similar loss functions on other U-Net based models making predictions on image data. Having this separated makes it easier to reference and keeps my other articles easier to understand.

This is based on the techniques demonstrated and taught in the Fastai deep learning course.

This loss function is partly based upon the research in the paper Losses for Real-Time Style Transfer and Super-Resolution and the improvements shown in the Fastai course (v3).

This paper focuses on feature losses (called perceptual loss in the paper). The research did not use a U-Net architecture as the machine learning community were not aware of them at that time.

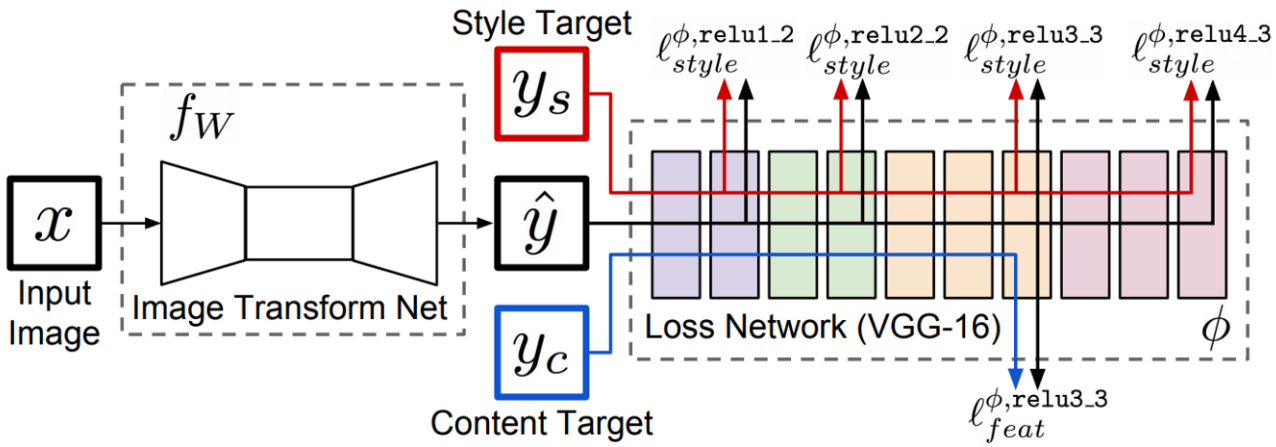


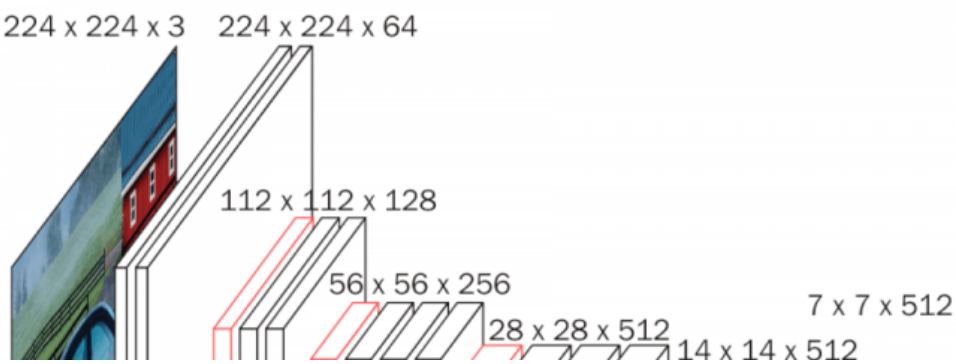
Fig. 2. System overview. We train an *image transformation network* to transform input images into output images. We use a *loss network* pretrained for image classification to define *perceptual loss functions* that measure perceptual differences in content and style between images. The loss network remains fixed during the training process.

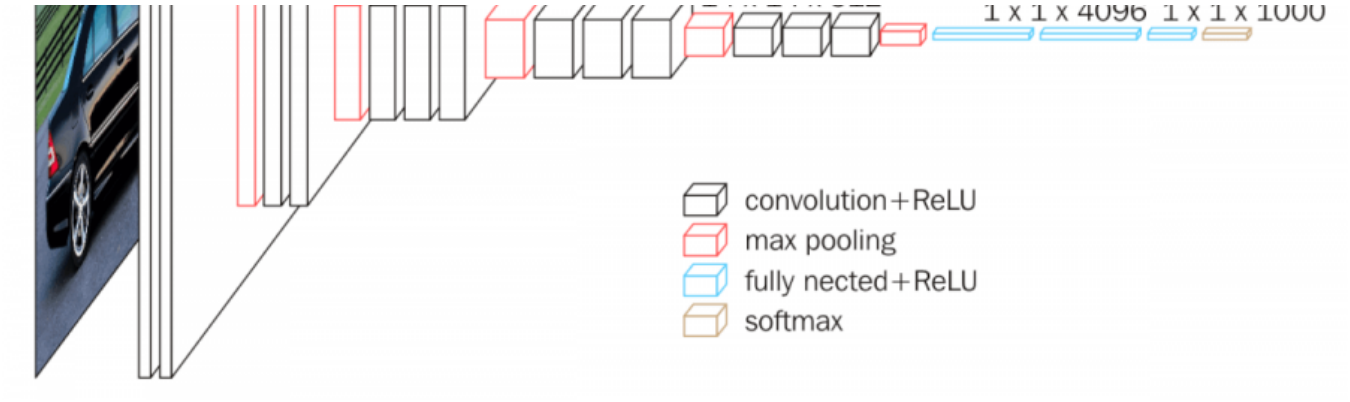
Source: Convolutional Neural Network (CNN) Perceptual Losses for Real-Time Style Transfer and Super-Resolution: https://arxiv.org/abs/1603.08155

This loss function used is similar to the loss function in the the paper, using VGG-16 but also combined with pixel mean squared error loss loss and gram matrix style loss. This has been found to be very effective by the Fastai team.

VGG-16

VGG is another Convolutional Neural Network (CNN) architecture devised in 2014, the 16 layer version is utilised in the loss function for training this model.

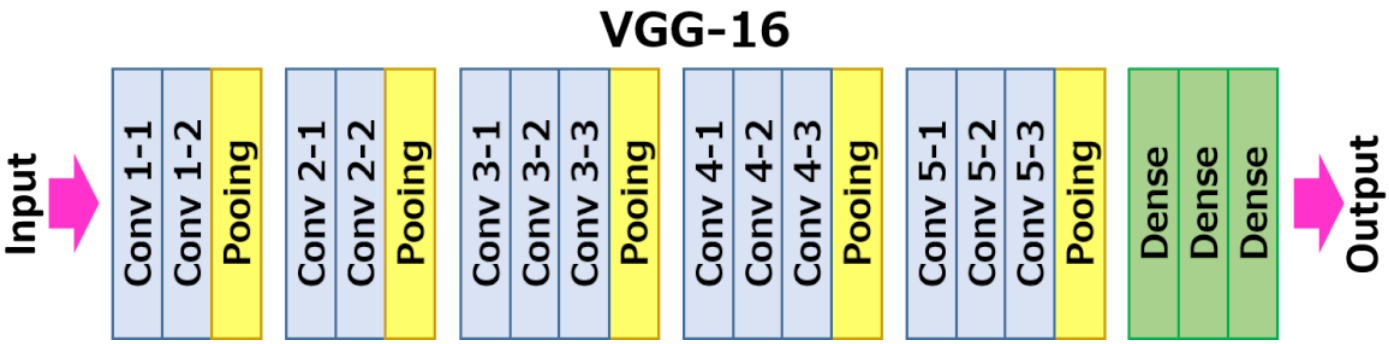




VGG-16 Network Architecture. Source: https://neurohive.io/wp-content/uploads/2018/11/vgg16-1-e1542731207177.png

The VGG model. a network pretrained on ImageNet, is used to evaluate the generator model's loss. Normally this would be used as a classifier to tell you what the image is, for example is this a person, a dog or cat.

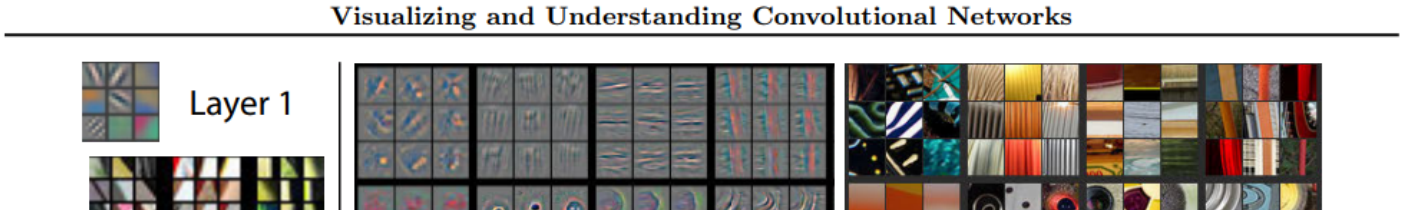
The head of the VGG model is the final layers shown as fully connected and softmax in the above diagram. This head is ignored and the loss function uses the intermediate activations in the backbone of the network, which represent the feature detections.



Different layers in VGG-16. Source: https://neurohive.io/wp-content/uploads/2018/11/vgg16.png

Those activations can be found by looking through the VGG model to find all the max pooling layers. These are where the grid size changes and features are detected.

Heatmaps visualising the layer activations for varied images can be seen in the image below. This shows examples of varied features detected in the different layers of network.



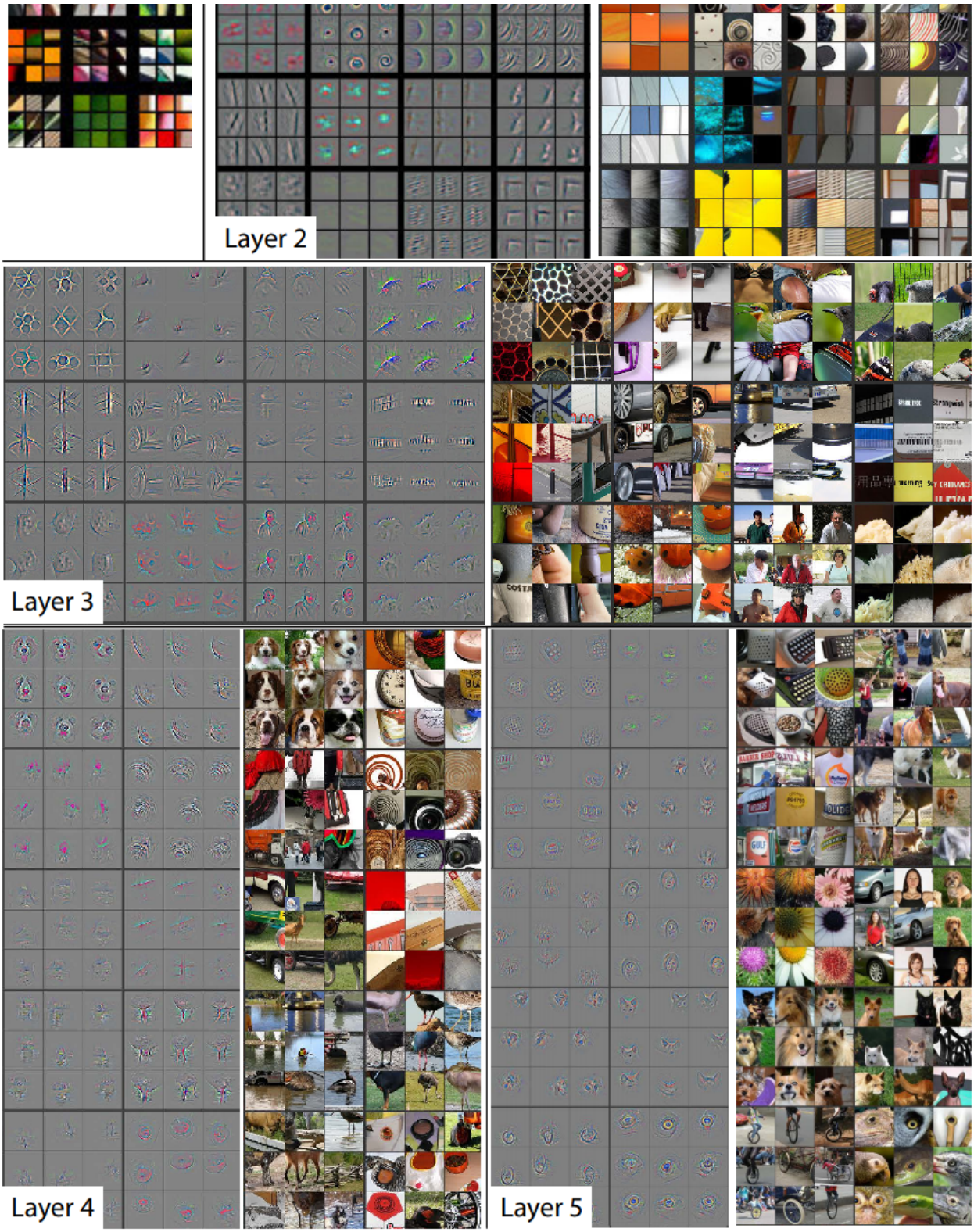


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form.

Visualisation of feature activations in CNNs. Source: page 4 of https://arxiv.org/pdf/1311.2901.pdf

The training of a model can use this loss function based on the VGG model's activations. The loss function remains fixed throughout the training unlike the critic

part of a GAN.

Feature losses

The Feature map has 256 channels by 28 by 28 which are used to detect features such fur, an eyeball, wings and the type material among many other type of features. The activations at the same layer for the (target) original image and the generated image are compared using mean squared error or the least absolute error (L1) error for the base loss. These are feature losses. This error function uses L1 error.

This allows the loss function to know what features are in the target ground truth image and to evaluate how well the model's prediction's features match these rather than only comparing pixel difference. This allows the model being trained with this loss function to produce much finer detail in the generated/predicted features and output.

Gram matrix style loss

A gram matrix defines a style with respect to specific content. By calculating the gram matrix for each feature activation in the target/ground truth image, it allows the style of that feature to be defined. If the same gram matrix is calculated from the activations of the predictions, the two can be compared to calculate how close the style of the feature prediction is to the target/ground truth image.

A gram matrix is the matrix multiplication of the each of the activations and the activation matrix's transpose.

This enables the model to learn and generate predictions of images whose features look correct in their style and in context, with the end result looking more convincing and appear closer or the same as the target/ground truth.

Predictions from models trained with this loss function

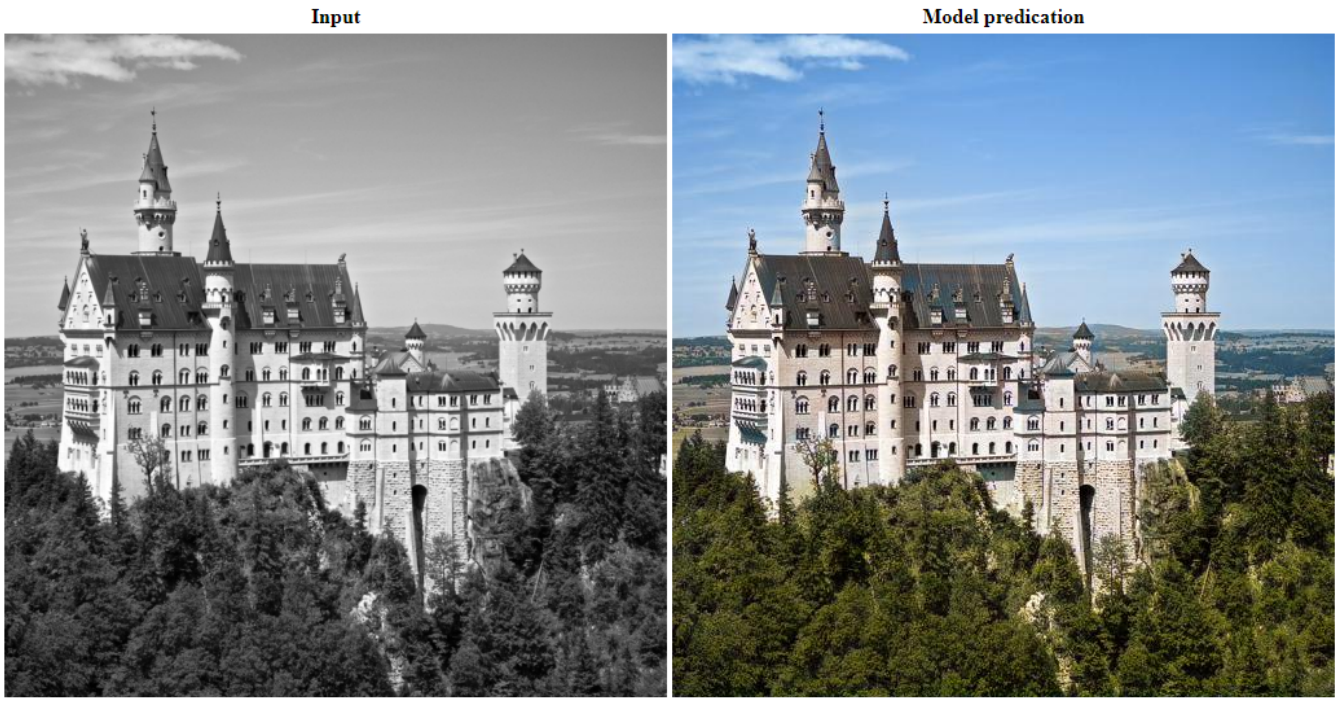
The generated predictions from trained models using loss functions based on these techniques have both convincing fine detail and style. That style and fine detail may be different aspects of image quality be predicting fine pixel detail or the predicting correct colours.

Two examples from models trained with loss functions based on this technique, showing how effective a model trained with this feature and style loss function can be:

From my super resolution experiments: https://towardsdatascience.com/deep-learning-based-super-resolution-without-using-a-gan-11c9bb5b6cd5

Super resolution on an image from the Div2K validation dataset

From my colourisation experiments, a link will be added when the article is published:



Enhancement of a Greyscale 1 channel image to a 3 channel colour image.

Fastai

Thank you to the Fastai team, without your courses and your software library I doubt I would have been able to learn about these techniques.

Machine Learning Deep Learning

About Help Legal

Get the Medium app